Title: SYSTEMS AND METHODS FOR CREATING NETWORK-BASED SOFTWARE SERVICES USING SOURCE CODE ANNOTATIONS
Docket No.: BEAS-01445US1 SRM/DTX Inventor(s): Marvin et al.
Appln. No.: Unknown Attorney: Sheldon R. Meyer
Confirm No: Unknown Phone: (415) 362-3800
Filed: Herewith Express Mail No.: EV 386447462 US

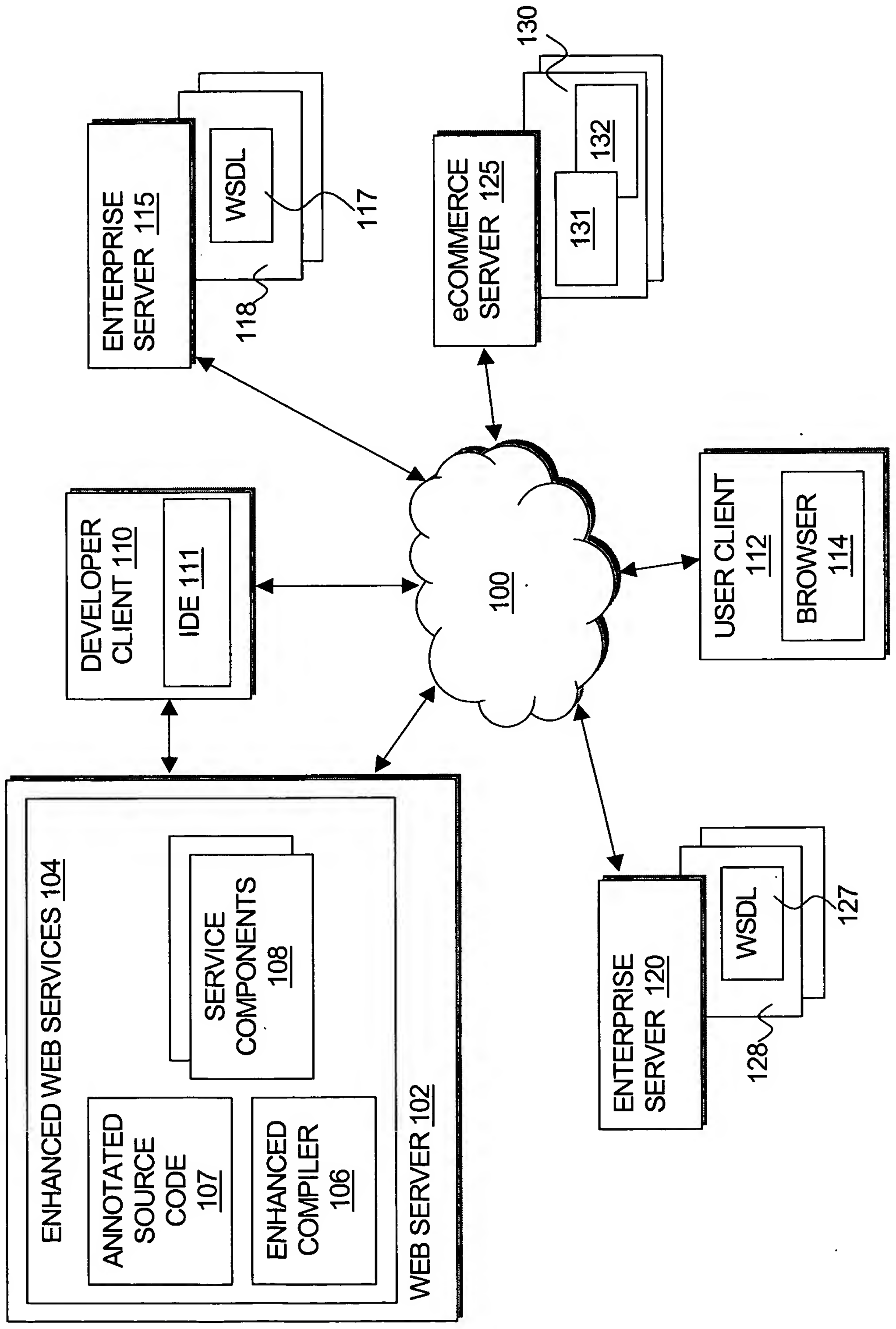
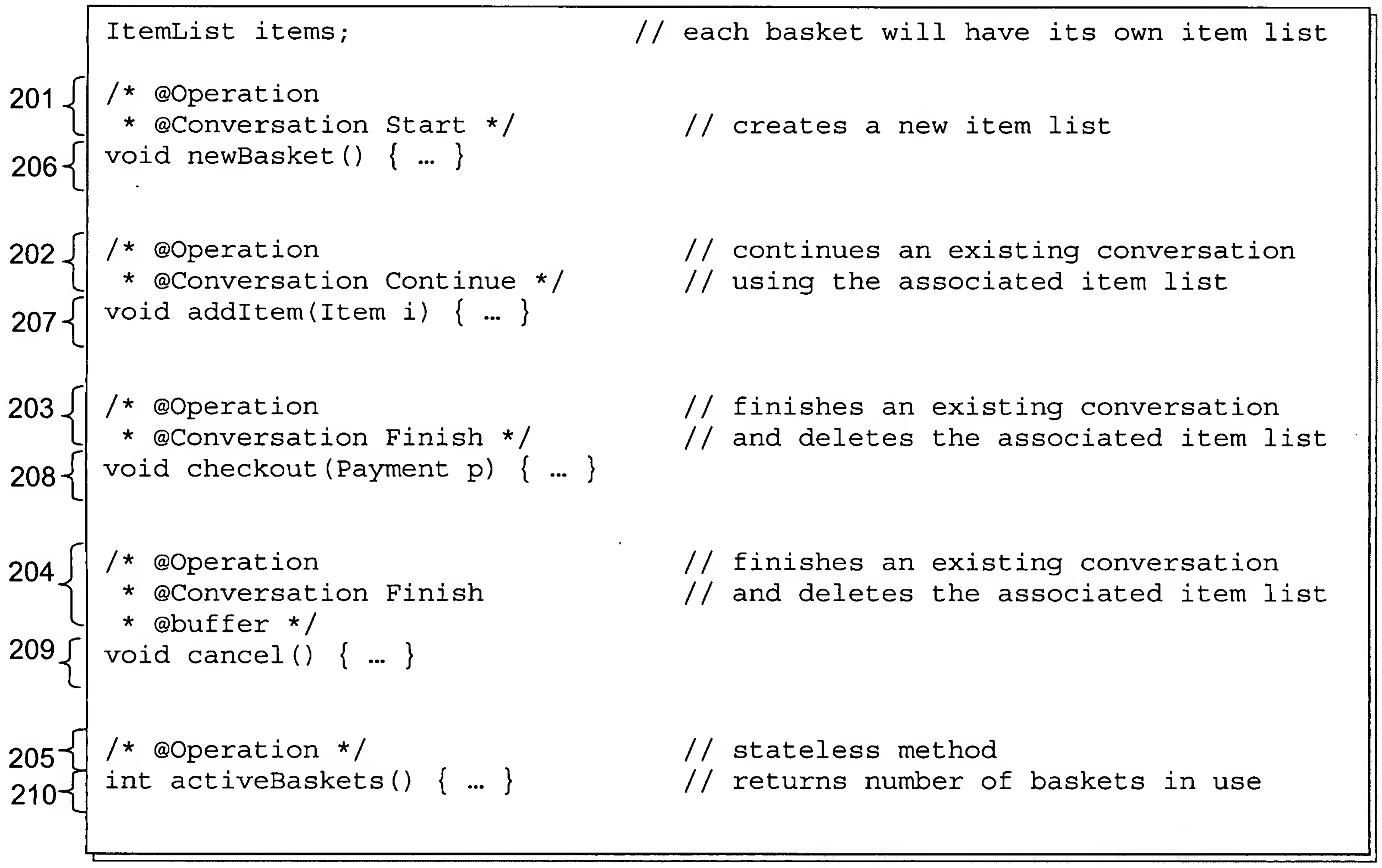


Figure 1



```
ItemList items;                    // each basket will have its own item list

/* @Operation
 * @Conversation Start */            // creates a new item list
void newBasket() { … }


/* @Operation                       // continues an existing conversation
 * @Conversation Continue */        // using the associated item list
void addItem(Item i) { … }


/* @Operation                       // finishes an existing conversation
 * @Conversation Finish */          // and deletes the associated item list
void checkout(Payment p) { … }


/* @Operation                       // finishes an existing conversation
 * @Conversation Finish             // and deletes the associated item list
 * @buffer */
void cancel() { … }


/* @Operation */                    // stateless method
int activeBaskets() { … }           // returns number of baskets in use
```

Figure 2

```
/**
  * @operation
   */
 public String greeting(String firstname, String lastname)
 {
   Calendar cal =
     Calendar.getInstance(TimeZone.getDefault());
   SimpleDateFormat sdf =
     new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

   return "Hello " + firstname + " " + lastname +
          " at " + sdf.format(cal.getTime());
 }
```

Figure 3A

```
POST /app/mypackage/CreditReport.jws HTTP/1.1
Host: www.company.com
Content-Type: text/xml; charset="utf-8"
Content-Length: 648
SOAPAction: "Some-URI"

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:cgc="http://bea.com/SOAP/conversation"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <soap:Header>
   </soap:Header>
   <soap:Body>
     <m:greeting xmlns:m="http://www.company.com/app/mypackage/CreditReport.jws">
       <firstname>Jane</firstname>
       <lastname>Doe</lastname>
     </m:greeting>
   </soap:Body>
</soap:Envelope>
```

Figure 3B

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: 574

<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <soap:Header>
   </soap:Header>
   <soap:Body>
      <m:greetingResponse
        xmlns:m="http://www.company.com/app/mypackage/CreditReport.jws">
          <result>Hello Jane Doe at 2001-08-15 16:18:04</result>
      </m:greetingResponse>
   </soap:Body>
</soap:Envelope>
```

Figure 3C

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:c="http://openuri.org/soap/conversation/">
  <s:Header>
    <c:conversationID>4232-133214-3223132:42332-3144-2332251</c:conversationID>
    <c:defaultCallbackURL>
      http://original.com/foo/Callbacks.jwi
    </c:defaultCallbackURL>
  </s:Header>
  <s:Body>
    <!-- contents omitted -->
  </s:Body>
</s:Envelope>
```

Figure 4

Title: SYSTEMS AND METHODS FOR CREATING NETWORK-BASED SOFTWARE SERVICES USING SOURCE CODE ANNOTATIONS
Docket No.: BEAS-01445US1 SRM/DTX Inventor(s): Marvin et al.
Appln. No.: Unknown Attorney: Sheldon R. Meyer
Confirm No: Unknown Phone: (415) 362-3800
Filed: Herewith Express Mail No.: EV 386447462 US
Sheet 5 of 15

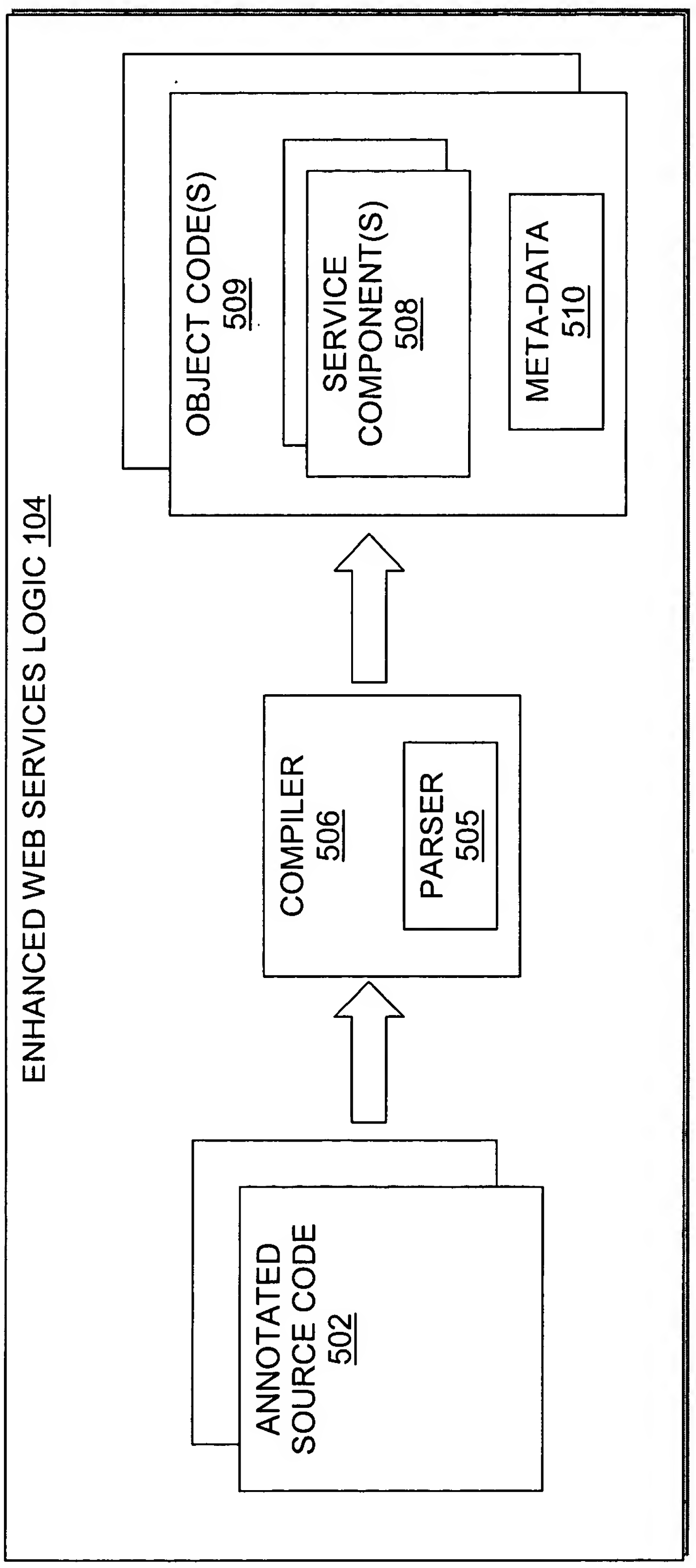


Figure 5

Title: SYSTEMS AND METHODS FOR CREATING NETWORK-BASED SOFTWARE SERVICES USING SOURCE CODE ANNOTATIONS
Docket No.: BEAS-01445US1 SRM/DTX Inventor(s): Marvin et al.
Appln. No.: Unknown Attorney: Sheldon R. Meyer
Confirm No: Unknown Phone: (415) 362-3800
Filed: Herewith Express Mail No.: EV 386447462 US

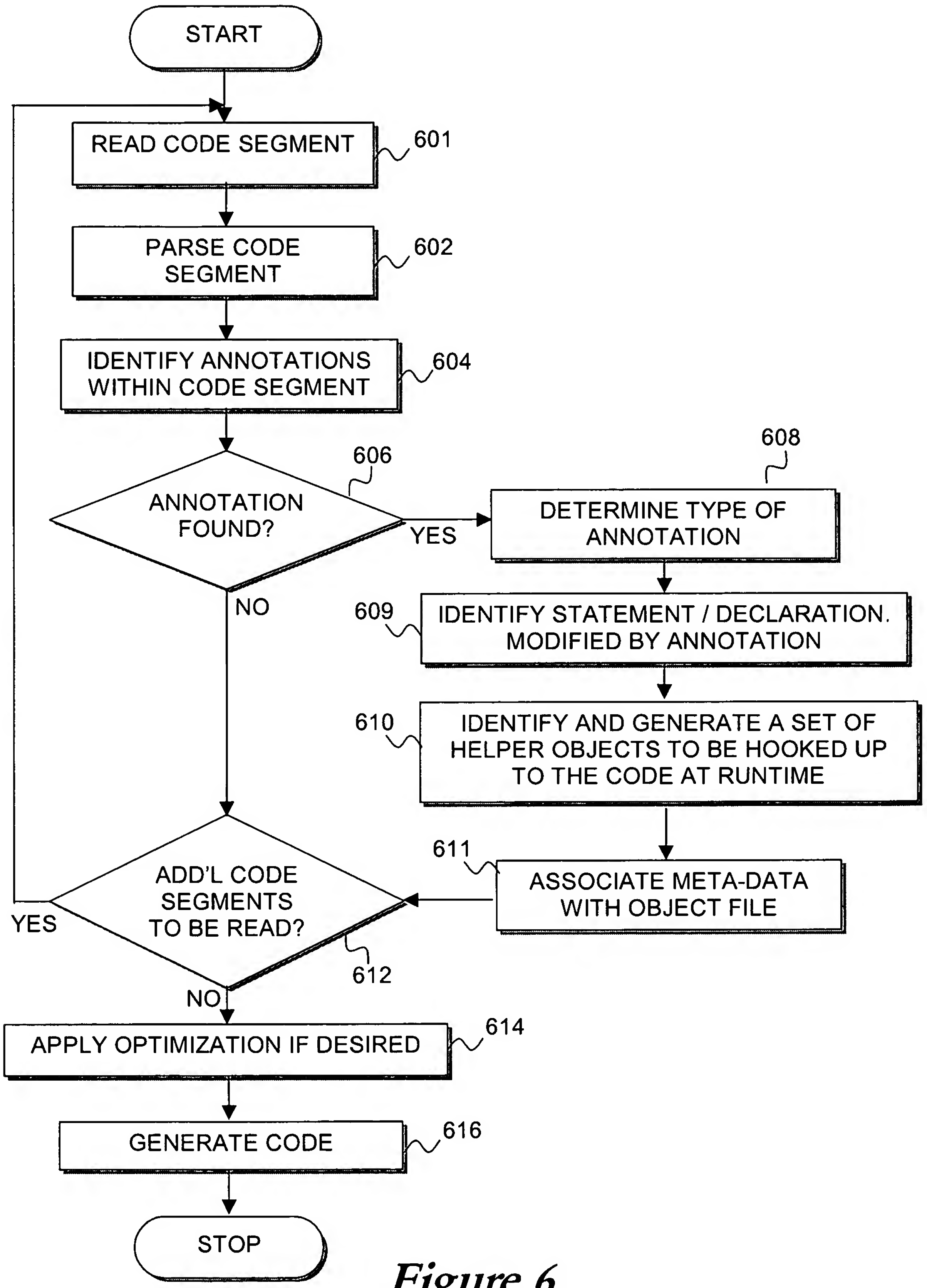

START
READ CODE SEGMENT
601
PARSE CODE SEGMENT
602
IDENTIFY ANNOTATIONS WITHIN CODE SEGMENT
604
ANNOTATION FOUND?
606
YES
NO
DETERMINE TYPE OF ANNOTATION
608
IDENTIFY STATEMENT / DECLARATION. MODIFIED BY ANNOTATION
609
IDENTIFY AND GENERATE A SET OF HELPER OBJECTS TO BE HOOKED UP TO THE CODE AT RUNTIME
610
ASSOCIATE META-DATA WITH OBJECT FILE
611
ADD'L CODE SEGMENTS TO BE READ?
612
YES
NO
APPLY OPTIMIZATION IF DESIRED
614
GENERATE CODE
616
STOP

Figure 6

Title: SYSTEMS AND METHODS FOR CREATING NETWORK-BASED SOFTWARE SERVICES USING SOURCE CODE ANNOTATIONS
Docket No.: BEAS-01445US1 SRM/DTX Inventor(s): Marvin et al.
Appln. No.: Unknown Attorney: Sheldon R. Meyer
Confirm No: Unknown Phone: (415) 362-3800
Filed: Herewith Express Mail No.: EV 386447462 US
Sheet 7 of 15

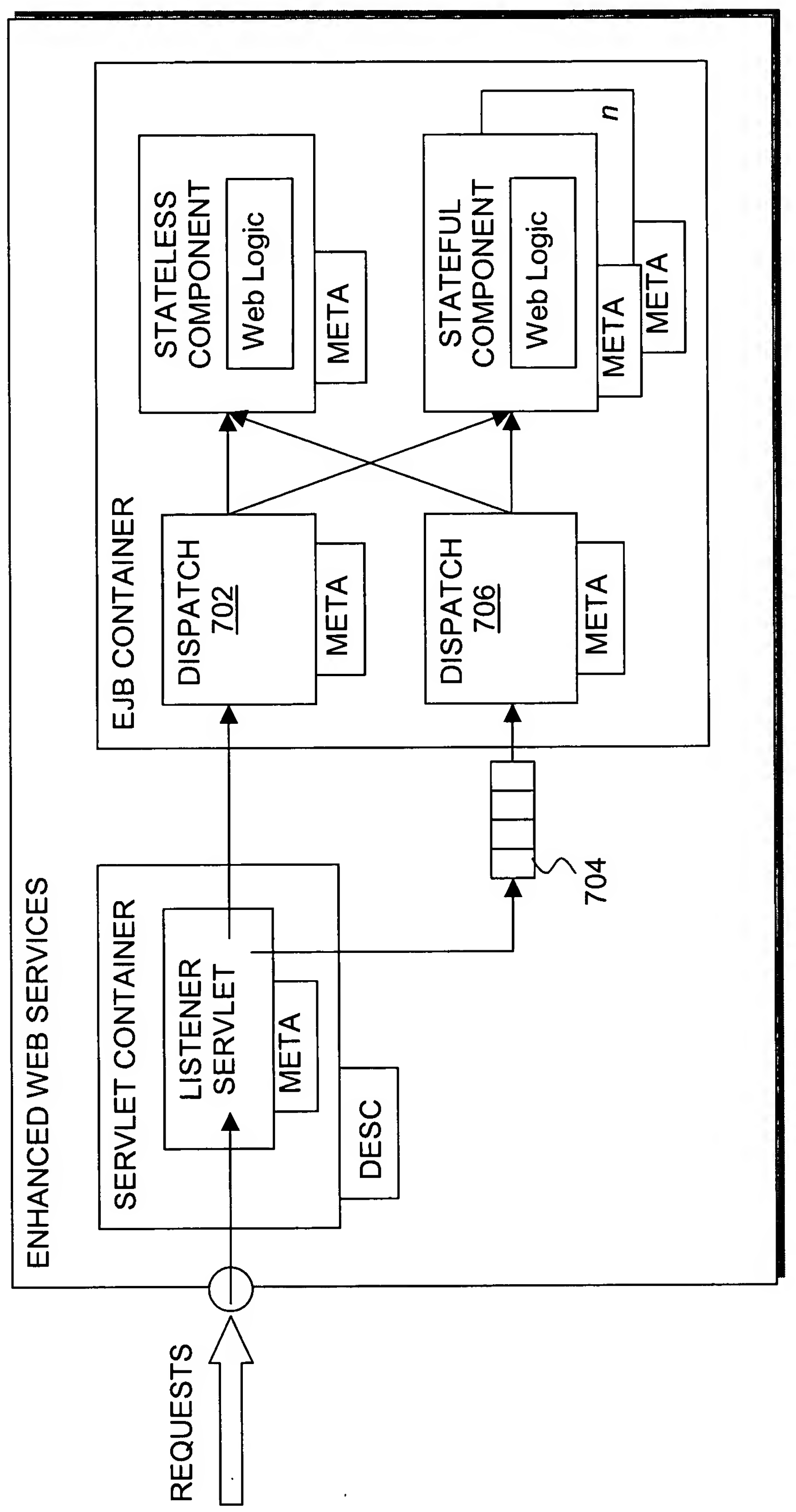


Figure 7

Title: SYSTEMS AND METHODS FOR CREATING NETWORK-BASED SOFTWARE SERVICES USING SOURCE CODE ANNOTATIONS
Docket No.: BEAS-01445US1 SRM/DTX Inventor(s): Marvin et al.
Appln. No.: Unknown Attorney: Sheldon R. Meyer
Confirm No: Unknown Phone: (415) 362-3800
Filed: Herewith Express Mail No.: EV 386447462 US

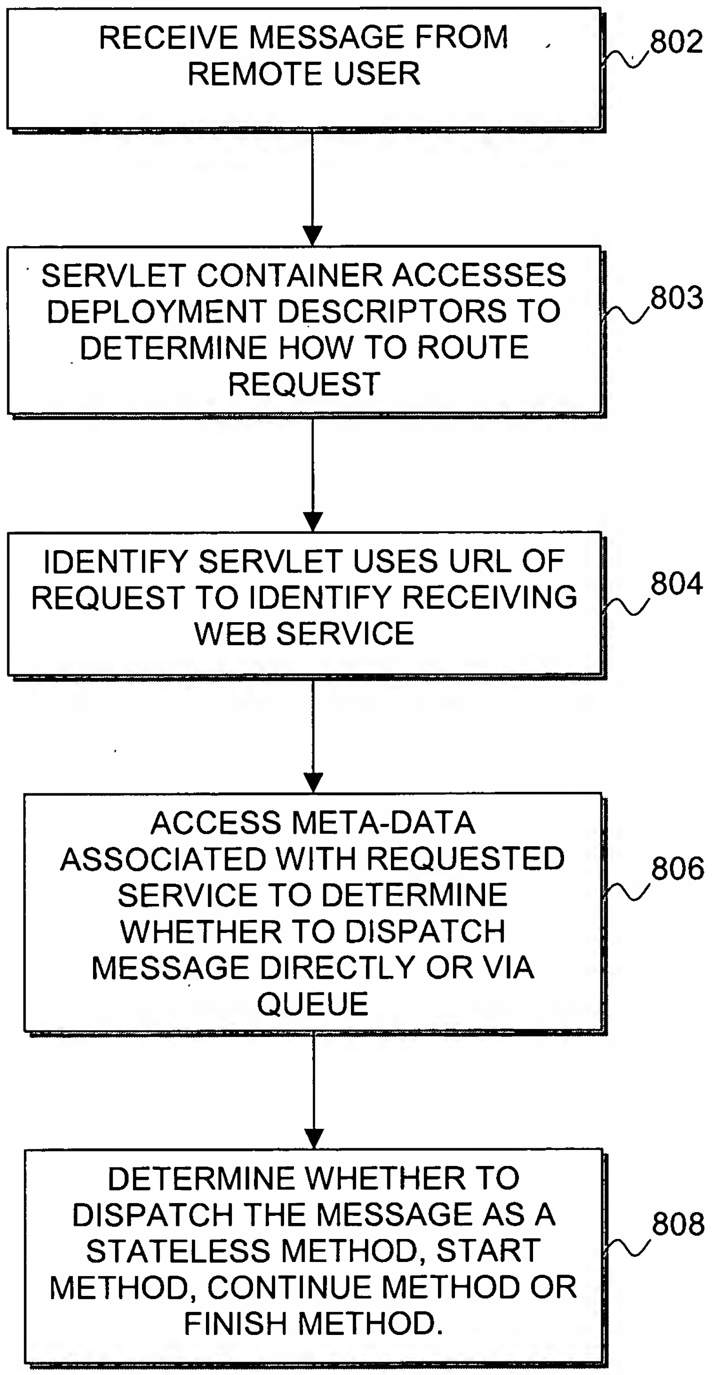

RECEIVE MESSAGE FROM REMOTE USER
802
SERVLET CONTAINER ACCESSES DEPLOYMENT DESCRIPTORS TO DETERMINE HOW TO ROUTE REQUEST
803
IDENTIFY SERVLET USES URL OF REQUEST TO IDENTIFY RECEIVING WEB SERVICE
804
ACCESS META-DATA ASSOCIATED WITH REQUESTED SERVICE TO DETERMINE WHETHER TO DISPATCH MESSAGE DIRECTLY OR VIA QUEUE
806
DETERMINE WHETHER TO DISPATCH THE MESSAGE AS A STATELESS METHOD, START METHOD, CONTINUE METHOD OR FINISH METHOD.
808

Figure 8

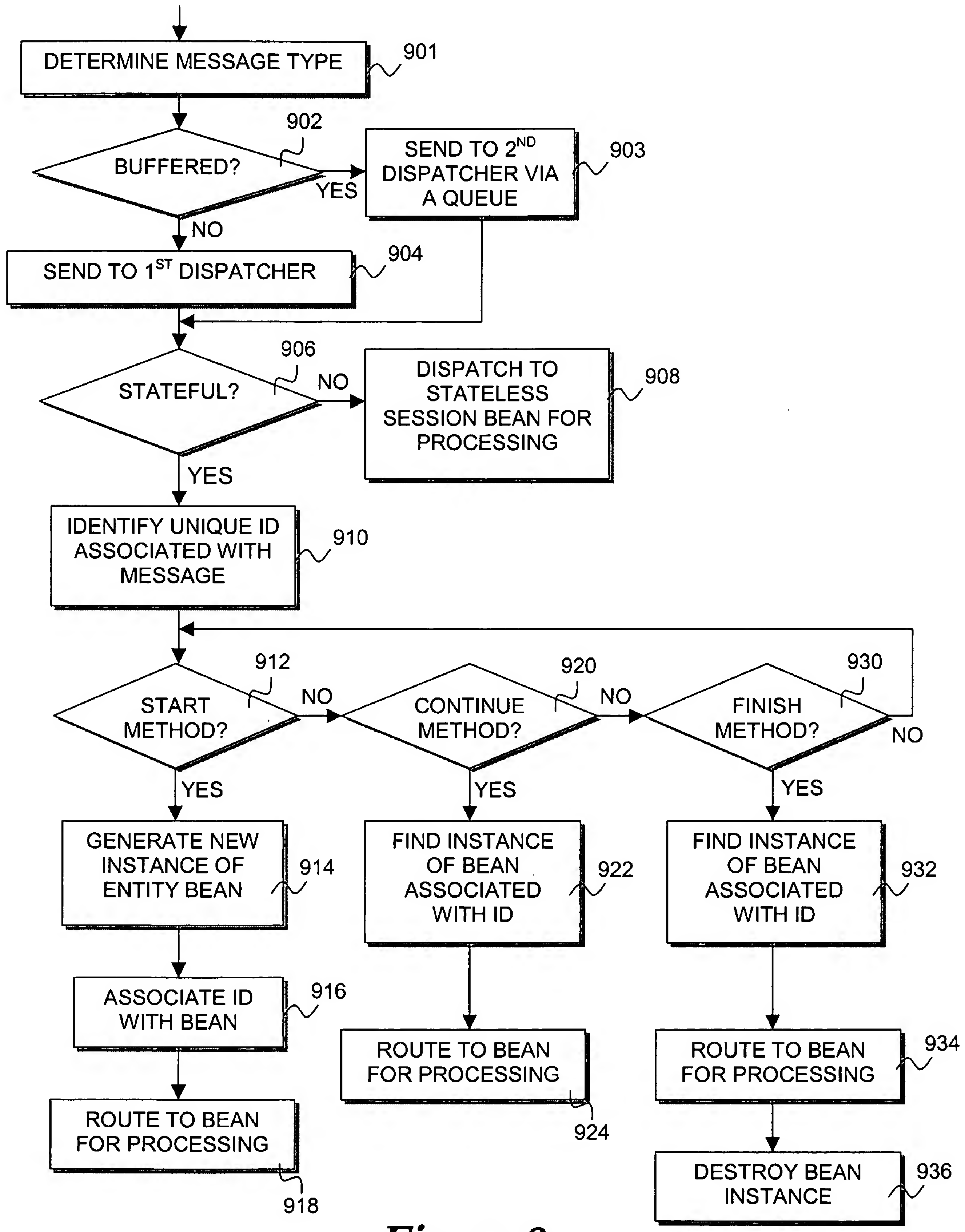

DETERMINE MESSAGE TYPE
901
902
BUFFERED?
YES
SEND TO 2ND DISPATCHER VIA A QUEUE
903
NO
SEND TO 1ST DISPATCHER
904
906
STATEFUL?
NO
DISPATCH TO STATELESS SESSION BEAN FOR PROCESSING
908
YES
IDENTIFY UNIQUE ID ASSOCIATED WITH MESSAGE
910
912
START METHOD?
NO
920
CONTINUE METHOD?
NO
930
FINISH METHOD?
NO
YES
GENERATE NEW INSTANCE OF ENTITY BEAN
914
ASSOCIATE ID WITH BEAN
916
ROUTE TO BEAN FOR PROCESSING
918
YES
FIND INSTANCE OF BEAN ASSOCIATED WITH ID
922
ROUTE TO BEAN FOR PROCESSING
924
YES
FIND INSTANCE OF BEAN ASSOCIATED WITH ID
932
ROUTE TO BEAN FOR PROCESSING
934
DESTROY BEAN INSTANCE
936

Figure 9

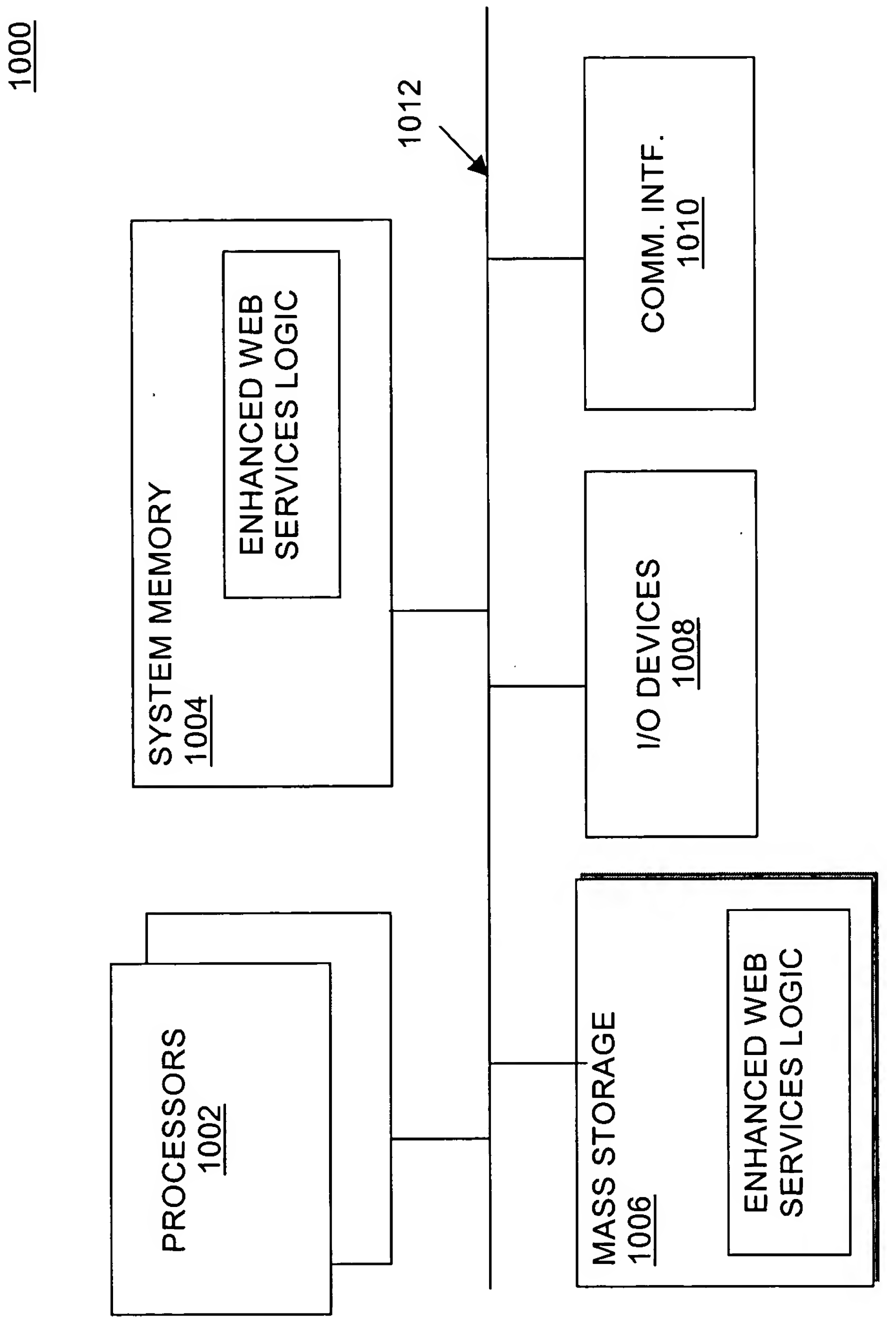
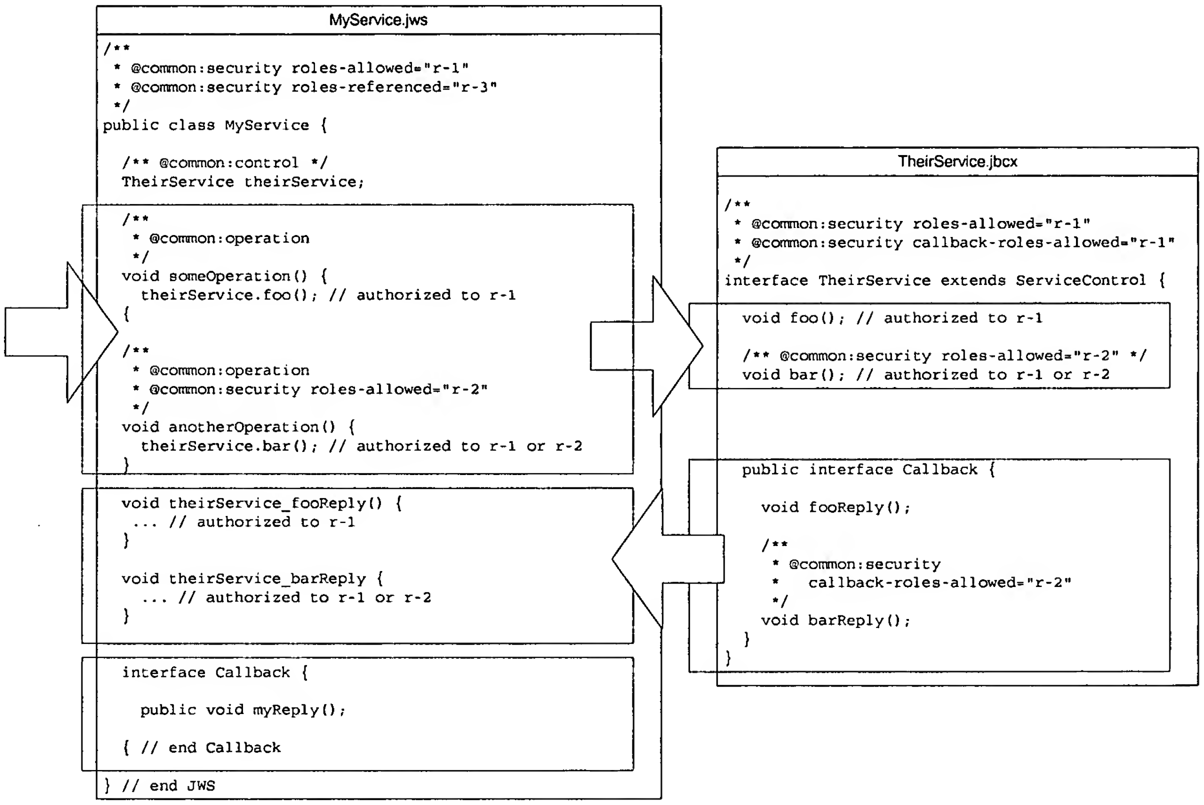

1000
1012
SYSTEM MEMORY 1004
ENHANCED WEB SERVICES LOGIC
PROCESSORS 1002
COMM. INTF. 1010
I/O DEVICES 1008
MASS STORAGE 1006
ENHANCED WEB SERVICES LOGIC

Figure 10



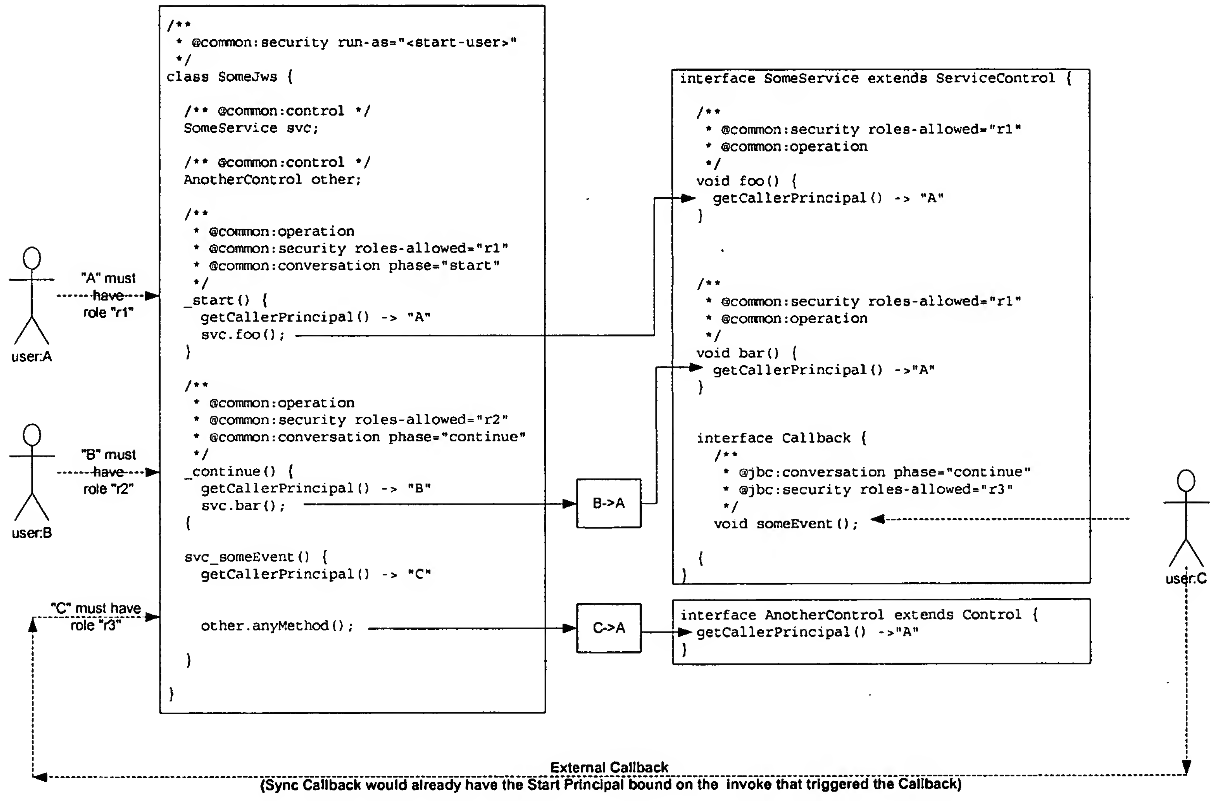
```
/**
 * @common:security roles-allowed="r-1"
 * @common:security roles-referenced="r-3"
 */
public class MyService {

  /** @common:control */
  TheirService theirService;

  /**
   * @common:operation
   */
  void someOperation() {
    theirService.foo(); // authorized to r-1
  {

  /**
   * @common:operation
   * @common:security roles-allowed="r-2"
   */
  void anotherOperation() {
    theirService.bar(); // authorized to r-1 or r-2
  }

  void theirService_fooReply() {
   ... // authorized to r-1
  }

  void theirService_barReply {
    ... // authorized to r-1 or r-2
  }

  interface Callback {

    public void myReply();

  { // end Callback
} // end JWS
```

```
/**
 * @common:security roles-allowed="r-1"
 * @common:security callback-roles-allowed="r-1"
 */
interface TheirService extends ServiceControl {
  void foo(); // authorized to r-1

  /** @common:security roles-allowed="r-2" */
  void bar(); // authorized to r-1 or r-2


  public interface Callback {

    void fooReply();

    /**
     * @common:security
     *   callback-roles-allowed="r-2"
     */
    void barReply();
  }
}
```

Figure 11

Using run-as="<start-user>" will cause the Subject that ***started*** the conversation to be bound to all continue|finish calls ***from*** the JW(X).

1. Invoke privileges on Control operations will be evaluated with respect to the roles granted to the starting Principal
2. From within called Controls:
 (a) Any call to getCallerPrincipal() will return the starting Principal
 (b) Any call to hadRole() will be evaluated with respect to the roles granted to the starting Principal



```
/**
 * @common:security run-as="<start-user>"
 */
class SomeJws {

  /** @common:control */
  SomeService svc;

  /** @common:control */
  AnotherControl other;

  /**
   * @common:operation
   * @common:security roles-allowed="r1"
   * @common:conversation phase="start"
   */
  _start() {
    getCallerPrincipal() -> "A"
    svc.foo();
  }

  /**
   * @common:operation
   * @common:security roles-allowed="r2"
   * @common:conversation phase="continue"
   */
  _continue() {
    getCallerPrincipal() -> "B"
    svc.bar();
  {

  svc_someEvent() {
    getCallerPrincipal() -> "C"

    other.anyMethod();

  }
}
```

```
interface SomeService extends ServiceControl {

  /**
   * @common:security roles-allowed="r1"
   * @common:operation
   */
  void foo() {
    getCallerPrincipal() -> "A"
  }

  /**
   * @common:security roles-allowed="r1"
   * @common:operation
   */
  void bar() {
    getCallerPrincipal() ->"A"
  }

  interface Callback {
    /**
     * @jbc:conversation phase="continue"
     * @jbc:security roles-allowed="r3"
     */
    void someEvent();

  {
}
```

```
interface AnotherControl extends Control {
  getCallerPrincipal() ->"A"
}
```

Figure 12

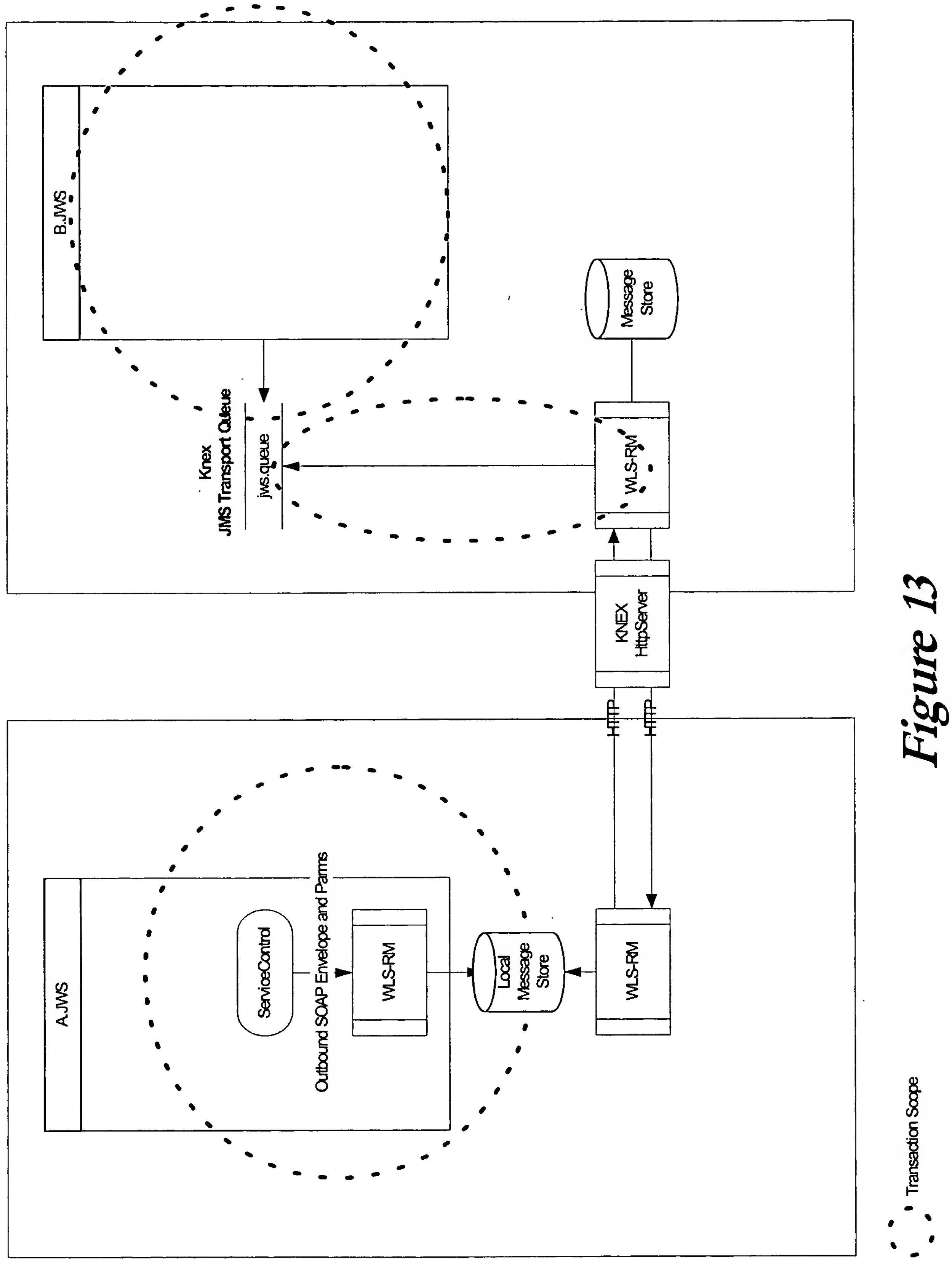

B.JWS
Knex
JMS Transport Queue
jws.queue
Message Store
WLS-RM
KNEX HttpServer
HTTP
HTTP
A.JWS
ServiceControl
Outbound SOAP Envelope and Parms
WLS-RM
Local Message Store
WLS-RM
Transaction Scope

Figure 13

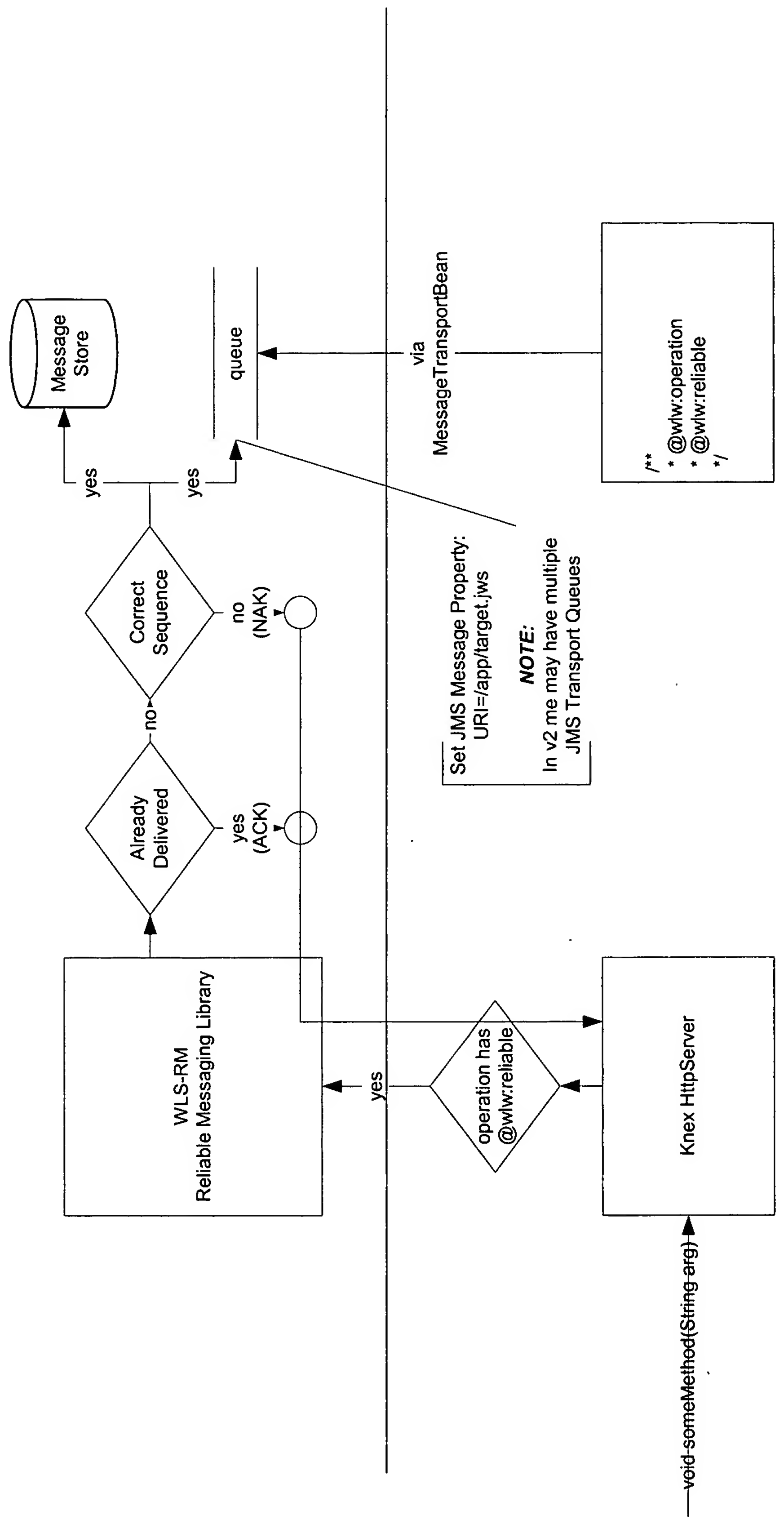

Message Store
queue
via MessageTransportBean
/**
* @wlw:operation
* @wlw:reliable
*/
yes
yes
Correct Sequence
no (NAK)
no
Already Delivered
yes (ACK)
Set JMS Message Property: URI=/app/target.jws
NOTE:
In v2 me may have multiple JMS Transport Queues
WLS-RM Reliable Messaging Library
yes
operation has @wlw:reliable
Knex HttpServer
void someMethod(String arg)

Figure 14

```
public interface ServiceInterceptor {

  /**
   * Provides initialization data to the interceptor implementation.  This method
   * will be called before any call to "handle" methods.
   * @param config configuration paramaters defined through config-data annotations
   * @param wsdl the definition for this service
   */
  public void init(Map config, XMLInputStream wsdl);

  /**
   * The interface called from the knex runtime to allow top-level
   * containers to interrogate and/or manipulate the incoming XML
   * request.  The request respresents the entire payload of the
   * request.  This interface is invoked before any paramter-xml
   * maps are run.
   * @param xmlRequest The request that is targeted to an @operation
   * method.
   * @return an XMLInputStream that will be delivered to an @operation
   * method on the current service.
   */
  public XMLInputStream handleRequest(XMLInputStream xmlRequest);

  /**
   * The interface called from the knex runtime to allow top-level
   * containers to interrogate and/or manipulate the outgoing XML
   * response.  The response respresents the entire payload of the
   * response. This interface is invoked after any return-xml
   * maps have run.
   * @param xmlResponse The response that was generated by an @operation
   * method.
   * @return an InputStream that will be returned to the caller of the
   * service.
   */
  public XMLInputStream handleResponse(XMLInputStream xmlResponse);

  /**
   * The interface called from the knex runtime to allow top-level
   * containers to interrogate and/or manipulate the outgoing XML
   * response.  The response respresents the entire payload of the
   * response.
   * @param xmlResponse The response that was generated by an @operation
   * method.
   * @return an InputStream that will be returned to the caller of the
   * service.
   */
  public XMLInputStream handleFault(XMLInputStream xmlFault);

  /**
   * Called when the service instance is being removed.  After destroy is called,
   * none of the "handle" methods will be calls.
   */
  public void destroy();
}
```

Figure 15